Mikko Maksimainen

Combining eye tracking and imaging luminance photometry in studying viewing conditions of night-time driving

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Helsinki 01.11.2013

Thesis supervisor:

Prof. Liisa Halonen

Thesis advisor:

D.Sc. Marjukka Puolakka

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Mikko Maksimainen

Title: Combining eye tracking and imaging luminance photometry in studying viewing conditions of night-time driving

Date: 01.11.2013 Language: English Number of pages:7+54

Professorship: Code:

Supervisor: Prof. Liisa Halonen

Advisor: D.Sc. Marjukka Puolakka

In this thesis an optimal experiment equipment to study night-time driving conditions is examined and clarified. The equipment should be able to gather information about luminance, subject's eye fixation and surrounding environment. The data gathered should be integratable and synchronizable. The main operative environment for the measurement system will be night-time traffic lighting measurement.

The optimal set of devices is examined by evaluating and comparing different eye tracking devices and imaging luminance photometers on the market. Also an initiative design for imaging luminance photometer with high sample rate is introduced. A rudimentary software for eye tracking data and luminance data integration is created. A possibility to integrate this data to mobile mapped environment models is discussed.

Keywords: lighting, outdoor lighting, eye tracking, mobile mapping systems, imaging luminance photometry

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Mikko Maksimainen

Työn nimi:

Päivämäärä: 01.11.2013 Kieli: Englanti Sivumäärä:7+54

Professuuri: Koodi:

Valvoja: .

Ohjaaja: .

Tässä diplomityössä selvitetään sopivaa mittauslaitteistoa valaistujen yöliikenneolosuhteiden mittaamiseen. Laitteiston tarkoitus on kerätä synkronisoitavissa ja yhdistettävissä olevaa tietoa luminanssista ja koehenkilön katseen kohdistumisesta suhteessa ympäröivään tilaan. Mittauslaitteiden pääasiallinen toimintaympäristö on öisen liikennevalaistuksen mittaustilanne.

Sopiva laitteisto etsitään arvioimalla ja vertailemalla markkinoilta löytyviä silmänseurantamittalaitteita ja kuvantavia luminanssimittareita. Alustava malli korkean näytteenottotaajuuden kuvantavasta luminanssimittarista esitellään. Luodaan myös ohjelma, joka pystyy yhdistämään silmänseurantatietoa luminanssitietoon. Myös mahdollisuutta yhdistää tätä informaatiota liikkuvasti kuvattuihin ympäristömalleihin pohditaan tässä diplomityössä.

Avainsanat: lighting, outdoor lighting, eye tracking, mobile mapping systems, imaging luminance photometry

Preface

Acknowledgements:
The authors acknowledge the Aalto Energy Efficiency research programme (project Light Energy - Efficient and Safe Traffic Environments) for financing this work.

I would like to thank everyone in the lighting unit and in the Light Energy project.

Helsinki, 01.11.2013

Mikko Maksimainen

Contents

Symbols and abbreviations

Symbols

L,Y	[cd/m^2]	Luminance (candela per square meter)
λ	[nm]	Wavelength

Abbreviations

ALS	Applied Sciences Laboratory
AR Marker	Augmented Reality Marker
CCD	Charge-Coupled Device
CCIR	Comité consultatif international pour la radio
CIE	Commission Internationale de l´Eclairage
CMOS	Complementary Metal Oxide Semiconductor
fps	Frames Per Second
GPS	Global Positioning System
HDR	High Dynamic Range
IMU	Inertia Measurement Unit
MMS	Mobile Mapping Systems
RGB	Red, Green and Blue
RYMM	Rakennetun Ympäristön Mittauksen ja Mallinnuksen Instituutti (The Institute for Measurement and Modelling of Built Environment)
SMI	Sensomotoric Instruments

1 Introduction and objectives

Outdoor lighting is an essential part of our living environment. With properly designed outdoor lighting we can improve safety and energy efficiency of our living surroundings.

Generally the safety of traffic increases when artificial lighting is installed [1]. Yet excessive lighting may cause light pollution and unnecessarily increased energy consumption. With contemporary technology the amount of light and the quality of light can be adjusted. But first it is important to have knowledge on where, when and how much illumination is needed.

The objective of this thesis is to find suitable equipment to measure traffic lighting conditions and human eye fixations in them. Such equipment include eye tracking device for eye fixation measurement and imaging luminance photometer for luminance measurement. The possibility of integrating this data into mobile mapped environment models is also examined.

The features and properties of different eye trackers and imaging luminance photometers available on the market was compared. Usability evaluation on these products was made either by testing the products or comparing the features mentioned in products' data sheets. The focus in the study was slightly emphasised on luminance photometers and their temporal performance. The ultimate goal was to find viable combinations of equipment suitable to gather compatible luminance and eye tracking data.

A core software performing the combination was also created. The software includes the basic functionality to combine the eye tracking data to pseudo-coloured luminance data video, but excludes the actual calibration and measurement reliability. The performance of the program created was evaluated and the programming interface for the future work was discussed. A demonstrative video of this integration is created and introduced. Future improvements and scalability for increased versatility were examined for the set of devices and software.

The market lacks an imaging photometer capable of high frequency data recording. Therefore an alternative not available on market was considered. Eye movement data synchronization to luminance data will benefit of high frequency luminance data. Thus an imaging luminance photometer that could capture luminance data at the same frame rate as the eye tracking devices capture eye tracking data (~ 30 Hz) was rudimentarily designed. A device for this alternative design was chosen and initial programming to achieve luminance data in this manner was piloted.

2 Background

2.1 Background theory on human vision

Human vision is based on photons between wavelengths of 400 nm to 750 nm gathered by the eye [2]. The photons gathered stimulate the receptor cells on the retina and the nerve stimuli are processed by brain to form an image. The photoreceptors are divided into two groups: cones used in photopic or color vision and rods used in scotopic vision. Photopic vision cone cells operate solely at adaptation luminance levels higher than $5\frac{\mathrm{cd}}{\mathrm{m}^2}$ and rods operate solely at adaptation luminance levels lower than $0.005\frac{\mathrm{cd}}{\mathrm{m}^2}$. The intermediate luminance region $0.005 - 5\frac{\mathrm{cd}}{\mathrm{m}^2}$ where both cones and rods operate simultaneously is called the mesopic region [3]. Photopic and scotopic spectral sensitivity functions are presented in Figure 1.

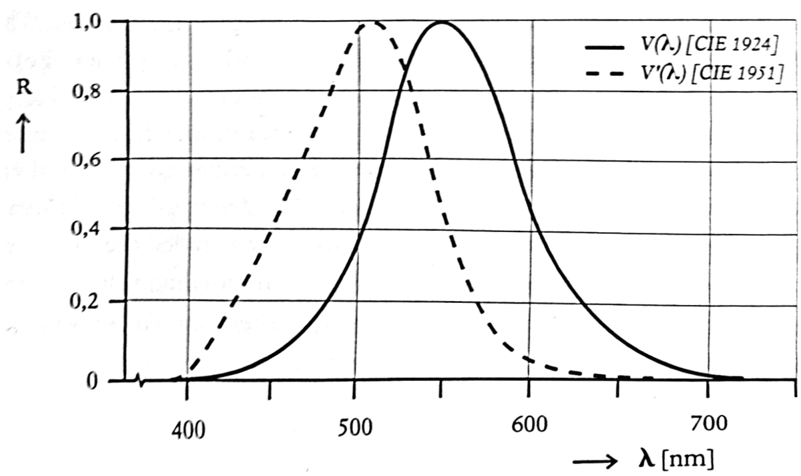

Figure 1. The relative photopic and scotopic spectral sensitivity functions. [2]

The CIE 191 is the system recommended in mesopic photometry [3]. CIE 191 recommends the method presented in Equation 1 to create mesopic spectral luminous efficiency function $V_{mes}(\lambda)$ using scotopic $V'(\lambda)$ and photopic $V(\lambda)$ spectral luminous efficiency functions [3].

$$M(m)V_{mes}(\lambda) = mV(\lambda) + (1 - m)V'(\lambda) \qquad \text{, for} \quad 0 \leq m \leq 1 \tag{1}$$

Where $M(m)$ is a normalizing function such that $V_{mes}(\lambda)$ attains a maximum value of 1. Here m is a coefficient the value of which depends on the visual adaptation conditions. If luminance level is purely photopic $m = 1$ and if purely scotopic $m = 0$. [3]

2.2 Background on visual field and eye movements

In order to effectively analyse and combine the gathered luminance data to eye movement data it is essential to understand the basics of human visual field and eye movements. The extend of foveal field is 2 °. The full binocular field of view is 180 ° horizontally and 126 ° vertically [4]. In Figure 2 the human binocular visual field is presented.

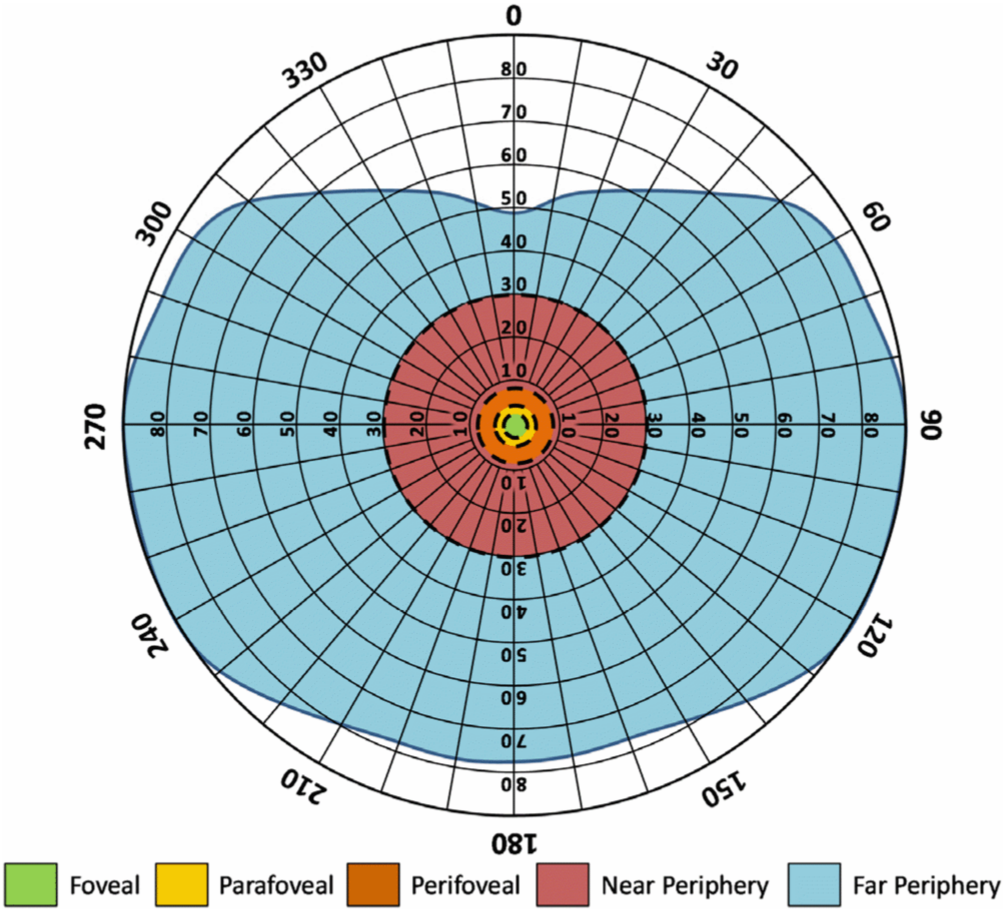

Figure 2. Human binocular visual field. [5]

Eye movements are separated to rapid saccadic movements and slower pursuit movements. Slow movements occur when eyes are following a moving object or when eyes are focusing on a point while the observer is slowly moving past the point. But when eyes scan the environment they move in rapid saccadic manner. Eyes fixate to a certain point for $0.2-0.6$ s until they fixate to the next target. Also even when eyes are fixated to a chosen point saccadic movements occur autonomously. [6]

3 Eye tracking equipment

3.1 Features of eye tracking equipment

Eye tracking equipment is a set of devices and software built to collect and analyse eye movement data. Mainly eye tracking focuses on the direction of the gaze, but also the changes in the pupil diameter or the amount and volume of saccades can be measured. A common way to present the gaze direction measurement is to combine video recordings of the eye movement to the respective video material on the visual scenery. [7]

Even though eye trackers can detect several aspects of human visual behaviour such as pupil dilation or blinking frequency, in this study the direction of gaze is the most relevant. In the following chapters eye tracking equipment from different manufacturers are introduced and compared. Also the analysis software integrated to the eye trackers are taken under evaluation. Finally the most suitable equipment for the task of studying human gaze direction under night-time driving conditions is suggested.

The important technical specifications for eye trackers are the following:

- Temporal resolution
- Spatial resolution
- Gaze tracking angles
- Scene camera resolution

Temporal resolution stands for the amount of measurements over certain time interval. Often temporal resolution is denounced as measurements per second [Hz] or frames per second (fps). The more measurements are made within a second the better. The physiology and functionality of human eye creates a systematic movement [6]. The fastest eye movements have a time interval of 20ms [8]. The eye movements occur with frequency of $\frac{1s}{0.020s} = 50$ Hz, but according to Nyquist criterion the sampling frequency must be more than double compared to the highest component of the information sampled to avoid folding. Therefore the eye tracking data recording frame rate should be > 100 Hz, if every single movement needs to be recorded. The second specification spatial resolution means the smallest angle difference that can be measured. The smaller the angle value the better the spatial accuracy is. Gaze tracking angles are the vertical and horizontal angles of the field measured. The wider the angles are the better. The fluctuations of a fixated eye can be as high as $0.5\,°$ with standard deviation of $0.2\,°$ [9]. Given that the full field of view is $180\,°$ horizontally and $126\,°$ vertically [4]. Thus a resolution of 900 horizontal and 630 vertical luminance measurement points should be sufficient.

Eye tracking equipment can roughly be divided to three categories in terms of their mount: head-sets, glasses and stationary devices. The head-sets and glasses are mobile and usable but demand a method to align the collected eye gaze data to the luminance data. Calibration tools such as augmented reality markers can be used to make data combination more accurate and less tedious [10]. Also the test subjects wearing glasses cannot wear eye tracking glasses simultaneously. Stationary eye-tracking devices can be more easily aligned with luminance measurements but require modifications to the automobile in use.

Scene camera is the part of an eye tracker that records the measurement field to a video file. The bigger the resolution and sample frequency values of the scene camera are the better. The video format of the recorded video should be standardized and common one to help the data post processing.

The data gathered is often presented either as heat maps or gaze traces [7]. Gaze traces are lines gaze draws to the scenery and heat maps show where in the scenery most of the sight concentrates over a time period. Both of the presentation styles are usable in driving behaviour measurements. For example SMI BeGaze software can do both of these presentations and also allows gathered data to be exported in organized form to be analysed in different software such as Matlab.

The market products reviewed are:

- Sensomotoric Instruments SMI Eye Tracking Glasses [11]
- Sensomotoric Instruments SMI iView X HED [12]
- SmartEye Pro 5.10 [13]
- Applied Science Laboratories ASL Mobile Eye XG [14]
- Tobii Glasses Eye Tracker [15]

3.1.1 Sensomotoric Instruments SMI Eye Tracking Glasses

Sensomotoric Instruments manufactures a variety of eye-tracking equipment and recording and analysis software. The equipment include glasses, a head-set and stationary devices. As in most of the devices available, the ones provided by Sensomotoric Instruments focus on market research.

SMI Eye Tracking Glasses are robustly built and usable eye tracking glasses. The device is equipped with a camera that captures the measured scenery forward and an infra-red camera that captures the eye movement. Technical details according to the SMI Eye Tracking Glasses data sheet are found in Table 1 and a picture of SMI Eye Tracking Glasses is presented in Figure 3.

Table 1. SMI Eye Tracking Glasses technical details. [11]

Temporal resolution	30 Hz
Spatial resolution	$0.1°$
Gaze tracking range	$80°$ horizontal, $60°$ vertical
Scene camera resolution	1280×960
Scene camera sample frequency	24 Hz
Scene camera video format	H.264

Figure 3. SMI Eye Tracking Glasses.

3.1.2 Sensomotoric Instruments or SMI iView X HED

Another product by Sensomotoric Instruments is iView X HED headset. It is a head mounted alternative for SMI Eye Tracking Glasses and, being head mounted, usable by test subject wearing glasses. Technical specifications from SMI iView X HED data sheet are found in following Table 2:

Table 2. SMI iView X HED technical details. [12]

Temporal resolution	$50 - 200$ Hz
Spatial accuracy	$0.1°$
Scene camera resolution	720×576
Scene camera video format	MPEG-4

The analysis and presentation software that accompanies SMI Eye Tracking Glasses is called BeGaze. BeGaze has intuitive and usable graphical user interface. The tracking data can be analysed with sampling frequencies up to 1250 Hz and parallel analysis among a group of test subjects is possible. Yet this high sampling frequency is not supported by the SMI eye-tracking devices reviewed. Data can be presented as heat maps and scan paths. BeGaze analysis software is capable to create usable and functional export files. The pupil dilatation and eye fixation data is separated clearly in individual columns. Figure 4 presents a heat map generated with Matlab using general example data exported from BeGaze. In presented heat map the accumulation of eye fixations increase changes the color in measured visual field. Blue color stands for no fixations and the redder the hue is the denser are the fixations. [11]

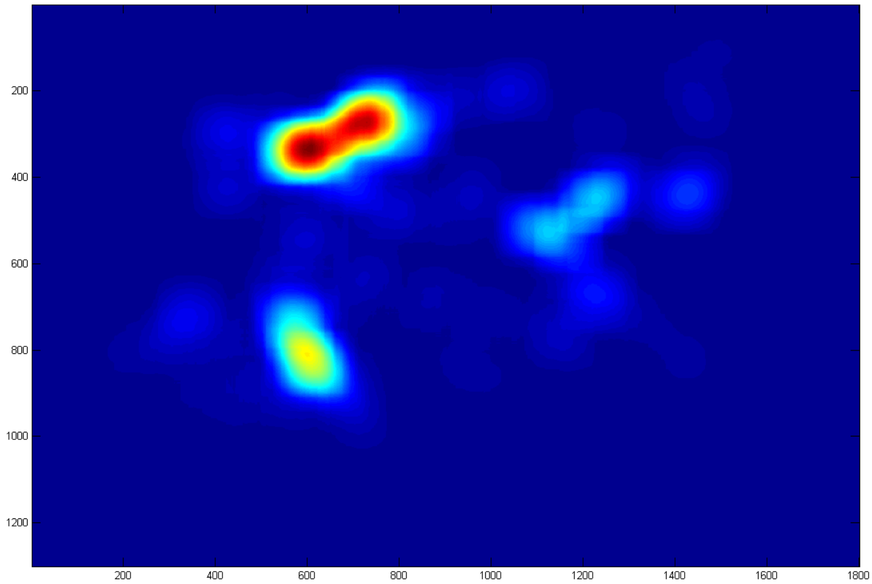

Figure 4. Demonstrative heat map presentation of eye tracking data.

3.1.3 SmartEye Pro 5.10

SmartEye is the only manufacturer that provides equipment designed especially to be mounted in a vehicle and then measure the eye movements of a driver. SmartEye Pro 5.10 equipment have a sampling rate of 60 Hz or 120 Hz. The 60 Hz sample rate is usable with up to 8 cameras and the 120 Hz sample rate is usable for up to 4 cameras. Their system is also scalable and can provide measurement angle of 360° when eight cameras are being used. The optimal functional distance between camera and a test subject is $30-300$ cm. With SmartEye Pro 5.10 one can measure head orientation, eye positions, gaze direction, pupil diameter, saccades, fixations, blinks and eyelid opening. A picture of SmartEye Pro 5.10 is shown in Figure 5 and its technical details are listed in Table 3.

Table 3. SmartEye technical details. [13]

Temporal resolution	$60 - 120$ Hz
Spatial accuracy	$0.5°$
Gaze tracking range	$90°$ - $360°$ horizontally and vertically

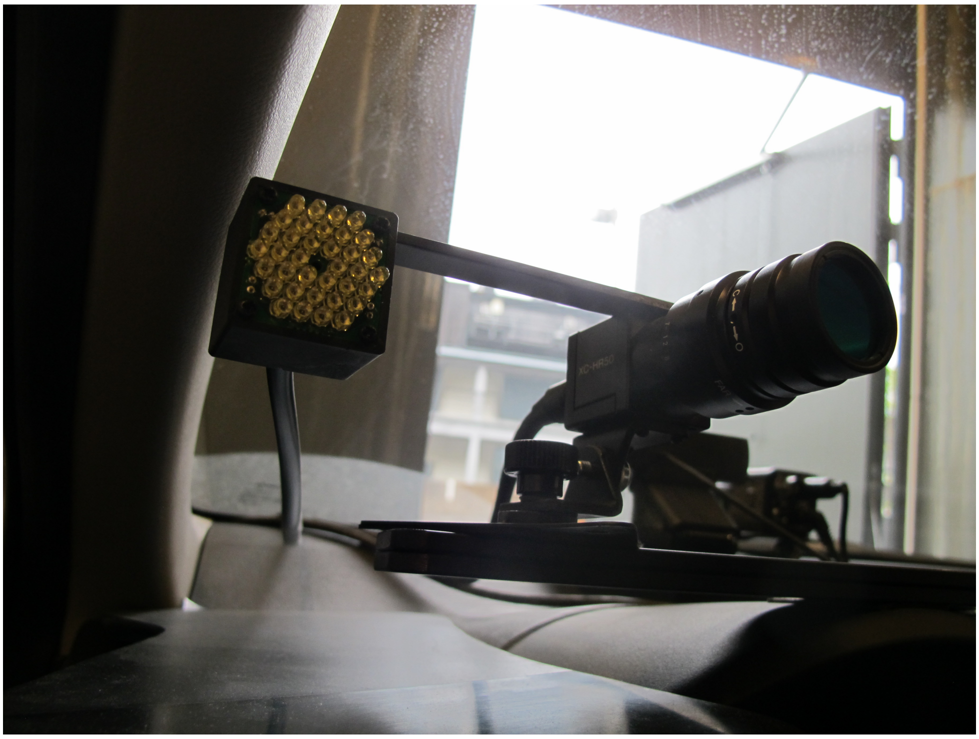

Figure 5. SmartEye eye tracking equipment installed in a research vehicle.

3.1.4 Applied Science Laboratories ASL Mobile Eye XG

ASL:s Mobile Eye XG is a product very similar to SMI Eye Tracking Glasses. It is designed to gather eye tracking data when researching consumer shopping behaviour, sports, reading and - most importantly in this study - driving. The important technical details are found in Table 4.

Table 4. ASL Mobile Eye XG technical details. [14]

Temporal resolution	30 Hz
Spatial accuracy	$0.5°$
Gaze tracking range	$50°$ horizontal, $40°$ vertical
Scene camera resolution	1600×1200

3.1.5 Tobii Glasses Eye Tracker

Tobii Glasses Eye Tracker is a product very similar to ASL and SMI glasses. The products presentation brochure only describes them as a tool for market research. The important technical details of Tobii Glasses Eye Tracker are listed in the following Table 5.

Table 5. Tobii Glasses Eye Tracker technical details. [15]

Temporal resolution	30 Hz
Gaze tracking angle	56° horizontal 40° vertical
Scene camera resolution	640×480
Scene camera sample frequency	30 Hz
Scene camera video format	MJPG2000

3.2 Comparison of eye tracking equipment

Reviewed eye tracking equipment are listed in rank order according to the specifications in Table 6.

The calibration and integration of eye tracking data to the scenery is very different in vehicle mounted equipment than in equipment worn by test subject [7]. Therefore equipment with different mount type are incomparable. The only vehicle mounted eye tracking equipment reviewed (SmartEye) requires a rather large investment both in money and work to construct a functional system. After constructing the system the research has to be done with that specific vehicle. Being vehicle mounted SmartEye Pro 5.10 provides the possibility to extend the measurement angle to up to 360° both horizontally and vertically by adding more eye tracking cameras to the measurement system.

According to chosen specifications the SMI Eye Tracking Glasses is the best alternative in eye trackers worn by test subjects, but only with a small margin. In stationary mounted devices SmartEye is trivially the best.

Table 6. Eye trackers in rank order according to the compared specifications

Criterion	1	2	3	4	5
Temporal resolution	SMI iView X HED (50Hz-200Hz)	SmartEye Pro 5.10 (60Hz-120Hz)	SMI Glasses, ASL Mobile Eye XG & Tobii Glasses (30Hz)	-	-
Spatial resolution	SMI Glasses, SMI iView X HED, SmartEye Pro 5.10, ASL Glasses (0.5°)	Tobii Glasses(no data)	-	-	-
Gaze tracking range	SmartEye Pro 5.10 (up to 360° horizontal and vertical)	SMI Eye Tracking Glasses (80° horizontal, 60° vertical)	Tobii Glasses (56° horizontal, 40° vertical)	ASL Glasses (50° horizontal, 40° vertical)	SMI iView X HED(no data)
Scene camera resolution	ASL Mobile Eye XG (1600 × 1200)	SMI Glasses (1280 × 960)	SMI iView X HED (720 × 576)	Tobii Glasses (640 × 480)	-

4 Imaging luminance photometers

Luminance photometer is a device used for measuring the luminance values of a visual field [16]. Luminance photometers can be divided to two types of devices: imaging luminance photometers and spot luminance photometers. With spot luminance photometer measurements are made from a small angle (for example $< 2°$ solid angle) and imaging luminance photometer measures luminance values of a wider scenery (up to $180°$ solid angle). When measuring a wide area the measurement time of imaging luminance photometer is much shorter compared to measuring the whole area with spot luminance meter.

An imaging luminance photometer is a digital camera implemented with either CCD or CMOS technology. In some cases the photometer even uses a digital camera available on the market. An example of this is the TechnoTeam LMK Mobile Advanced which uses Canon EOS 350D digital camera [17]. The important part of imaging photometry is the software that corrects the errors of the optics such as vignetting and optical distortion and then interprets the raw image as absolute luminance values. The whole system (hardware and software) is calibrated using known luminance surfaces to be able to interpret images as luminance maps.

In this chapter the different hardware and software products on the market are introduced and compared, keeping in mind the goal of combining eye tracking system with luminance photometry to study drivers lighting and viewing conditions under night-time traffic conditions. Using a sterling digital camera as an imaging luminance photometer is examined and methods of interpreting the taken images as matrices of luminance values is also taken under analysis and development.

The market products reviewed are:

- TechnoTeam LMK Mobile Advanced [17]
- TechnoTeam LMK5 Video [18]
- Radiant Zemax ProMetric 1400 [19]
- Instrument Systems LumiCam 1300 [20]

4.1 Features of imaging photometers

The quality and functionality of different imaging photometers can be evaluated by following criteria:

- Resolution of luminance image
- Dynamic resolution
- Dynamic resolution HDR

- Measurement time
- Calibration uncertainty
- Variety of optics
- Optimal working conditions

Resolution of luminance image means the amount of luminance values – horizontally and vertically – the photometer is able to present. Dynamic resolution of a single image is the relative difference in luminance levels the device is able to present. If dynamic resolution is 1:1000 and the highest measured luminance is $1000\frac{\mathrm{cd}}{\mathrm{m}^2}$ then the smallest luminance the photometer is able to present is $1\frac{cd}{m^2}$. Hence to measure the whole mesopic region ($0.005 - 5\frac{\mathrm{cd}}{\mathrm{m}^2}$), a dynamic resolution of 1:1000 is needed. To measure wider luminance region, for example $0.001 - 1000\frac{\mathrm{cd}}{\mathrm{m}^2}$, a dynamic resolution of 1:1000000 is needed. The dynamic resolution of HDR image means the extended dynamic resolution achieved by combining multiple images taken with different exposure times. Measurement time is the time needed to complete a measurement. It is desirable to minimize the measurement time. Calibration uncertainty is the relative error limit of each measurement pixel in percentage. It is desirable for an imaging luminance photometer to be supported with large variety of optics to make the device more versatile. Also it is desirable for an imaging luminance photometer to be usable in various conditions such as sub-zero temperatures or humid environment.

4.1.1 TechnoTeam LMK Mobile Advanced

LMK Mobile Advanced is a mobile luminance photometer, based on Canon EOS 350D digital camera. The manufacturer has calibrated the camera and optics to generate absolute luminance values out of the images taken with TechnoTeam LabSoft analysis software. The result is a mobile luminance measuring system.

LMK Mobile Advanced is a practical imaging luminance photometer. Its dynamic resolution is 1:4000 for a single image and 1:32000 for HDR image. It can be used with either versatile $18 - 50$ mm lens or with ultra-wide angle 8 mm fisheye lens. The lens mentioned first is suitable for traffic lighting luminance photometry and the fisheye lens is practical in room measurements where a viewing angle of $180\,^{\circ}$ is needed to achieve wider measurement field. The LMK Mobile camera itself is easy and intuitive to use. Being based on an open market digital camera it has standardized digital camera usability and tripod compatibility.

The LabSoft software is used to analyse and present the images taken with the LMK Mobile Advanced as luminance images. An example of a luminance measurement presented with LabSoft is in Figure 6. HDR presentations are created in LabSoft by combining 2 or 3 images. Even though LabSoft has professional tools for analysis its poor user interface makes it almost unusable. Data analysis using this software

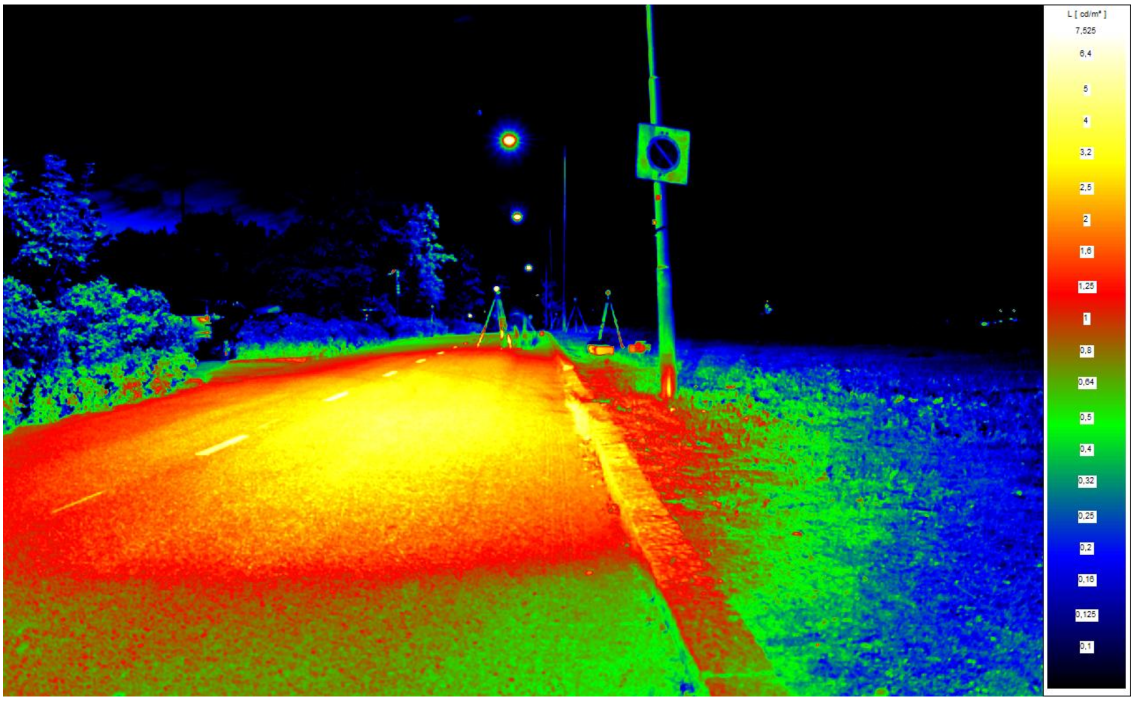

Figure 6. LabSoft luminance image.

is slow and inflexible. [17]

LMK Mobile is a useful tool when luminance values of a static area are measured. Yet LMK Mobile can not be used to collect continuous and simultaneous data with eye tracking equipment because it is incapable to record video. Also the measuring time for a single image under mesopic conditions are up to 10 or 15 seconds. Hence real time dynamic measurements are completely impossible.

LMK Mobile Advanced technical specifications are found in Table 7.

Table 7. LMK Mobile Advanced technical details. [17]

Resolution of the luminance image	2136 × 1424
Dynamic resolution, single	1 : 4000
Dynamic resolution, HDR	1 : 32000
Lenses	18 − 50mm and 8mm
Measurement time	0.25ms − 30s
Calibration uncertainty	2.5%
Sensor	CMOS Canon APS-C

4.1.2 TechnoTeam LMK5 Video

As LMK Mobile Advanced, LMK5 Video is also product of German manufacturer TechnoTeam. Where LMK Mobile Advanced is based on Canon EOS 350D digital camera, is LMK5 a custom made product. LMK5 uses Sony ICX CCD censor cells to capture the image in resolution from 1380×1030 up to 4008×4008 pixels. With single picture dynamic resolution is $1 : 1100$ and with composition $1 : 3600$. With high dynamic range (HDR) measurement the dynamic resolution is $1 : 1000000$. The spectral matching is done following the $V(\lambda)$ -function but for example $V'(\lambda)$ -function matching filter is also available. For measuring luminance levels of $0.01 \frac{\mathrm{cd}}{\mathrm{m}^2}$ integration time is $2-3$ seconds, but measurement is possible in 0.2 seconds if 10% error is acceptable. Yet 0.2 second measurement time only means a frame rate of 5 fps. Table 8 shows the technical details of LMK5 Video. [18]

Table 8. LMK5 Video technical details. [18]

Resolution	1380×1030
Dynamic resolution single	1:1100
Dynamic resolution HDR	$1 : 100000$
Exposure time	0.2ms – 30s
Calibration uncertainty	2.5% – 10%
Sensor	CCD Sony ICX 285 AL

4.1.3 Radiant Imaging ProMetric Color 1400

ProMetric 1400 is a purpose-built imaging luminance photometer not based on a digital camera existing on market. It uses a 14 bit 3072×2048 pixel charge-coupled device (CCD) as a sensor. A measurement time for full frame color image (3072×2048) is $25s$, but cropped images of 1536×1024 and 768×512 pixels can be measured in $8s$ and $5.5s$ respectively. Dynamic resolution of Prometric Color 1400 is $1 : 16384$ (14bit). The technical specifications of ProMetric 1400 are listed in Table 9.

Table 9. Radiant Imaging ProMetric Color 1400 technical details. [19]

Resolution of luminance image	3072×2048
Dynamic resolution of single image	$1 : 16384$
Measurement time	$5.5 - 25s$
Operation temperature	$0 - 30$ °C
Operation humidity	20 – 70% non-condensing
Calibration uncertainty	3%
Sensor	CCD

4.1.4 Instrument Systems LumiCam 1300

Instrument Systems LumiCam 1300 is also a purpose-built imaging luminance photometer. It is a viable alternative for LMK Mobile Advanced. Its weak points are the same as in LMK Mobile Advanced i.e the long measurement time. The construction of LumiCam 1300 is analogical to a digital camera. The CCD sensor used has 1280×1000 pixels. The measurement time of LumiCam 1300 varies from 1ms to 30s and at $10\frac{\mathrm{cd}}{\mathrm{m}^2}$ it is 0.7s. The Dynamic range of LumiCam 1300 is 1:4600 for single image and 1:6000000 for HDR image. LumiCam 1300 is capable of measuring luminance levels from $0.0001\frac{\mathrm{cd}}{\mathrm{m}^2}$ to $100000\frac{\mathrm{cd}}{\mathrm{m}^2}$. The widest lens (28mm) provides only a $22\,^\circ$ measurement field of view. The demanded operation temperature of LumiCam 1300 is $10-30\,^\circ\mathrm{C}$ which will certainly be problematic during the winter when night-time traffic conditions often occur in Finland. The technical specifications of LumiCam 1300 are listed in Table 10.

Table 10. Instrument Systems LumiCam 1300 technical details. [20]

Sensor	CCD
Resolution	1280×1000
Dynamic resolution single	1 : 4600
Dynamic resolution HDR	1 : 6000000
Dynamic range	$0.0001\frac{\mathrm{cd}}{\mathrm{m}^2}$ to $100000\frac{\mathrm{cd}}{\mathrm{m}^2}$
Lenses	28 mm and 50 mm and 105 mm
Measurement time	$0.001-30$ s
Operation temperature	$10-30\,^\circ\mathrm{C}$
Operation humidity	max. 70% non-condensing
Calibration uncertainty	4%

4.2 Comparison of imaging luminance photometers

Reviewed imaging luminance photometers are listed in rank order according to the important specifications in Table 11.

The products reviewed diverse in their rank orders depending on the specification scrutinized. ProMetric Color 1400 has the best values in resolution and dynamic resolution of single image, but lacks HDR capability and performs worst in measurement time. Lumicam 1300 has superior value in HDR image dynamic resolution, but has worst spatial resolution. LMK Mobile Advanced and LMK5 perform best in measurement time, but do not excel in other criteria. Market lacks a product where target in design is fast measuring frequency with little less emphasis on the accuracy. For conventional luminance measurements any of the compared devices is usable, for high sample rate data recording, none of them is.

Table 11. Imaging Luminance Photometers in rank order according to the compared specifications.

Criterion	1	2	3	4
Resolution of luminance image (pixels)	ProMetric 1400 (3072×2048)	LMK Mobile Adv. (2136×1424)	LMK5 (1380×1030)	LumiCam 1300 (1280×1000)
Dynamic resolution, single image	ProMetric 1400 ($1 : 16348$)	LumiCam 1300 ($1 : 4600$)	LMK Mobile Adv. ($1 : 4000$)	LMK5 ($1 : 1100$)
Dynamic resolution, HDR image	LumiCam 1300 ($1 : 6000000$)	LMK5 ($1 : 100000$)	LMK Mobile Adv. ($1 : 32000$)	ProMetric 1400 (no HDR)
Measurement time	LMK5 ($0.0002s - 30s$)	LMK Mobile Adv. ($0.00025s - 30s$)	LumiCam 1300 ($0.001s - 30s$)	ProMetric 1400 ($5.5s - 25s$)
Calibration uncertainty	LMK5 & LMK Mobile Adv. ($2.5 - 10\%$)	ProMetric 1400 (3%)	LumiCam 1300 (3%)	-

5 Digital video camera used for imaging luminance photometry

None of the imaging luminance photometers are fully suitable to be combined with the eye tracking equipment. The weakness of all the products reviewed is the long measurement time and the lack of possibility to measure real-time luminance at the same rate eye trackers can record the eye tracking data. This difficulty was faced for example in a study conducted as a combined effort of Aalto University and University of Helsinki to integrate eye tracking and luminance data in night time driving [21].

Therefore a development of a sterling digital camera being used as a high sample rate imaging luminance photometer was started. A supporting data analysis and result presenting software environment was also designed.

Together with Measuring and Modelling Institute for Built Environment of Aalto University a Nikon D800e digital camera was chosen to be the device for a new imaging luminance photometer system. In a pilot experiment images were taken with two cameras (the Nikon D800e and LMK Mobile Advanced) in a laboratory under mesopic luminance level conditions. Later both cameras were used in outdoor experiments where also video was taken with Nikon D800e. Using the calibrated data exported from TechnoTeam LMK LabSoft analysis software it was possible to fit the Nikon D800e image to an analogue luminance curve and thus create a calibration for the device.

In the following section the characterisation and rudimentary calibration of Nikon D800e is made. The conclusions of this study should give assets to convert also other advanced cameras e.g. Canon 5D to be applied as luminance measuring device. The future aspects including the possibilities and difficulties of the process are discussed also.

5.1 Nikon D800e features

Table 12. Nikon D800e technical details. [22]

Sensor	CMOS
Resolution	7360×4912
Video resolution	1920×1080 @ 30 fps
Dynamic resolution	1 : 16384
Codec	H.264/MPEG-4
Operation temperature	$0 - 40$ °C

Nikon D800e is a professional digital camera with 36.3 million effective pixel CMOS sensor. It is capable to shoot video with size of 1920×1080 pixels at 30 frames per second and size 1280×720 at 60 frames per second. The video compression used is H.264/MPEG-4 Advanced Video Coding. Any Nikon F mount lens can be mounted to this camera. Nikon specifications are listed in Table 12. [22]

5.2 The use of Nikon D800e as an imaging luminance photometer

Nikon D800e is a digital camera comparable to the Canon EOS 350D used in LMK Mobile Advanced imaging luminance photometer. Nikon D800e is superior digital camera to Canon EOS 350D in terms of technical specifications. In Canon EOS 350E the total amount of pixels is 7.96 millions and in Nikon D800e the amount of pixels is 36.3 millions. The main problem is that it will take a lot of work before the images taken by Nikon D800e can be fully and reliably interpreted as a matrix of luminance values. Even more difficult it is to create environment that gives reliable luminance values for video data. Accuracy decrease is almost inevitable because in video the amount of data easily grows vastly and thus compression is a necessity.

5.2.1 Laboratory and field experiments on Nikon D800e

The first step in Nikon D800e calibration was to take images with it and LMK Mobile Advanced in laboratory. The target of the images taken was a gray scale printed on paper. The luminance level of the white area in the object was set to $10\frac{\mathrm{cd}}{\mathrm{m}^2}$, $1\frac{\mathrm{cd}}{\mathrm{m}^2}$ and $0.1\frac{\mathrm{cd}}{\mathrm{m}^2}$ by the luminaire control in the laboratory room. The luminance value of the white area was measured with spot luminance meter (LMT L 1009). Later these experiments will be reproduced with more controlled environment to reduce the error. Figure 7 presents the experiment set up.

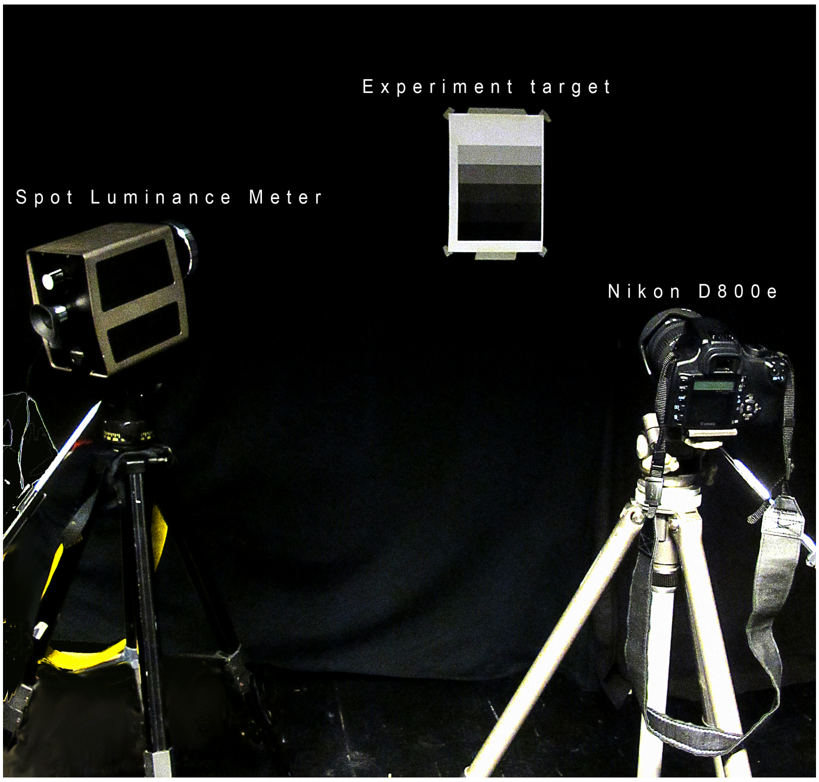

Figure 7. Laboratory Experiment Set Up.

The second experiment took place outdoors during night-time at Otaniemi, Espoo. The luminance values of the street Otakaari were measured. The street Otakaari was chosen because it was already mobile mapped and modelled by Built Environment Measurement and Modelling Institute of Aalto University (RYMM). Still-photography measurements of the chosen road were done with LMK Mobile Advanced and Nikon D800e. Then Nikon D800e was used to gather video data with following arrangements: car driving upfront, stable camera; car passing by, stable camera; camera mounted on the car driving along the road. A situational picture of the field experiment can be found in Figure 8, and LMK Mobile Advanced luminance image taken during the experiment in Figure 9.

Figure 8. Situational Picture of the Field Experiment (Copyright: Juho-Pekka Virtanen).

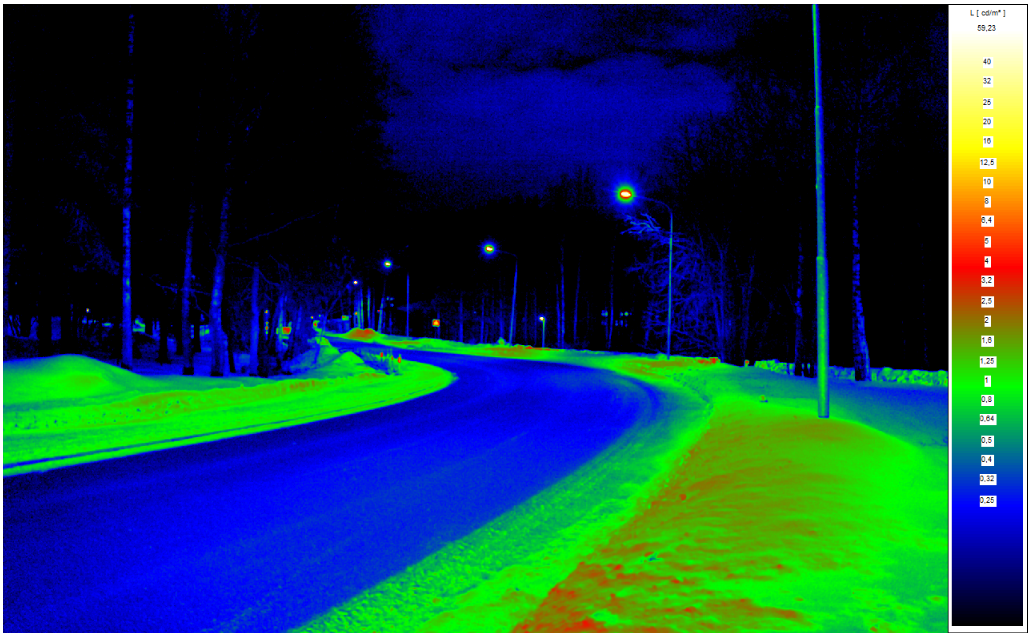

Figure 9. LMK Mobile Advanced Luminance Image from the Field Experiment.

5.2.2 Data analysis

To analyse the data the programming and data presenting tool Matlab was chosen. The LMK Mobile Advanced data was naturally first analysed with TechnoTeam's own software LabSoft. The matrix of luminance values was then exported from LabSoft and imported to Matlab to be used as a reference and comparative to Nikon D800e images. The Nikon D800e measurements images were imported to Matlab and the red, green and blue (RGB) values of the raw image was initially transformed to luminance values using the following formula:

$$Y = 0.3 * R + 0.59 * G + 0.11 * B \tag{2}$$

The formula used was chosen following the CCIR 601 standard of digital formats [23]. The more accurate coefficients introduced in the original standard were rounded to simplify the calculations. The original values are 0.299, 0.587, 0.114. Simplification of calculations is reasonable because on image is composed of over 36 million pixels and each second of video includes 30 images. Therefore the number of numerical calculations is very large. The formula of the standard reflects the $V(\lambda)$ function of standard observer introduced in CIE 1931 [24].

5.2.3 Data calibration

The actual knowledge how the luminance values are calibrated in the case of LMK Mobile Advanced is an industry secret. Thus achieving this knowledge is impossible. Therefore function was empirically fitted and fine adjusted to find correlation between the Nikon D800e luminance image and the analysed data from LMK Mobile Advanced.

The empirically fitted function used in data calibration for each value of the pixel matrix is the following:

$$Y_c = (1.016^{Y_s}) \tag{3}$$

Where Y_c stands for calibrated luminance and Y_s stands for the luminance value converted from the RGB values of the raw image. The values Y_c are then normalized by multiplying the Y_c matrix with m:

$$Y_{c\_normalized} = mY_c \qquad , m = (\frac{Y_l}{Y_m}) \tag{4}$$

Where Y_l is the maximum luminance value of luminance value matrix generated with LabSoft and Y_m is the highest value of the luminance value matrix under calibration. Then the low luminance values ($< 0.05\frac{cd}{m^2}$) are fitted with the following function:

$$Y_{c\_low} = (Y_c^{1.016}) * 0.58 \qquad , \text{for } Y_c < 0.05 \tag{5}$$

Thus we achieve the piecewise created fitting for the luminance value matrix of Nikon D800e. The full Matlab code is presented in Appendix A. The comments in the code specify the function of each separate part of it.

5.2.4 Improvements to D800e calibration and characterization

In the future the characterization studies for both LMK Mobile Advanced as the reference photometer and Nikon D800e as the chosen development model should be reproduced with high scientific standards. The vignetting and optical distortion of Nikon D800e and the optics chosen should be characterized and calibrated. The Matlab program will need optimization to be able to convert an image to a matrix of luminance values with good reliability.

5.2.5 Nikon D800e luminance data video presentation

The Matlab code (Appendix A) created also contains rudimentary functions for luminance data video presentation. It is clear that efficiency optimization is utterly important when handling larger data entities. Therefore a use of Python or C languages should be considered to implement the video processing. The optimum data compression and the synchronization marking were left for the future study.

5.2.6 Results of Nikon D800e data analysis

The results of data analysis are presented with Figures 10, 11 and 12. Figure 10 presents the luminance values measured with LMK Mobile Advanced and analysed by the TechnoTeam LabSoft analysis software. Figures 11 and 12 present the luminance values analysed with Matlab script using the LMK Mobile Advanced image and the Nikon D800e image respectively.

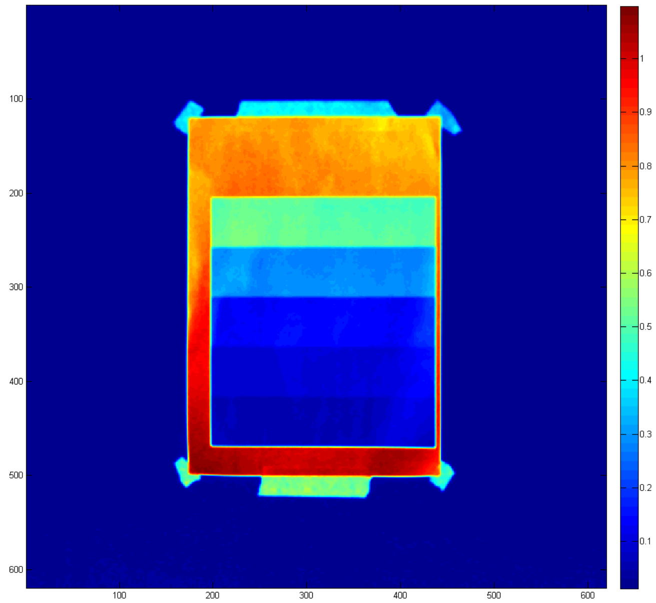

Figure 10. LMK Mobile Advanced Luminance Values Exported from LabSoft $(0 - 1.018\frac{\mathrm{cd}}{\mathrm{m}^2})$.

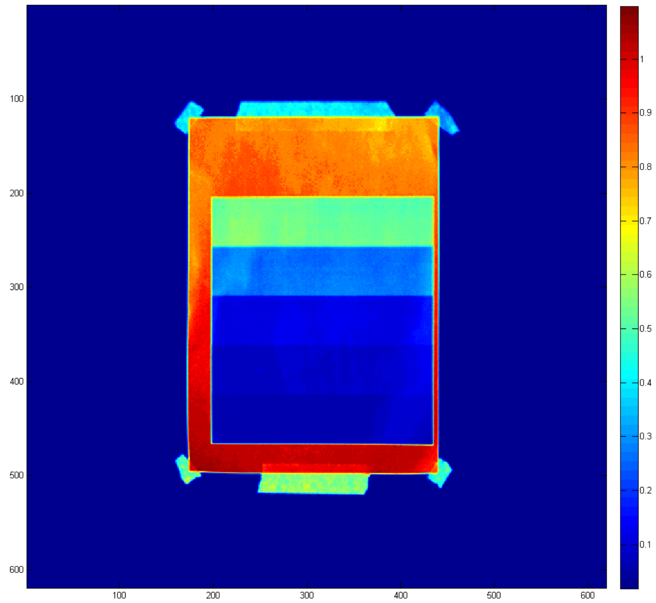

Figure 11. LMK Mobile Advanced Luminance Values Exported from Matlab $(0 - 1.018\frac{\mathrm{cd}}{\mathrm{m}^2})$.

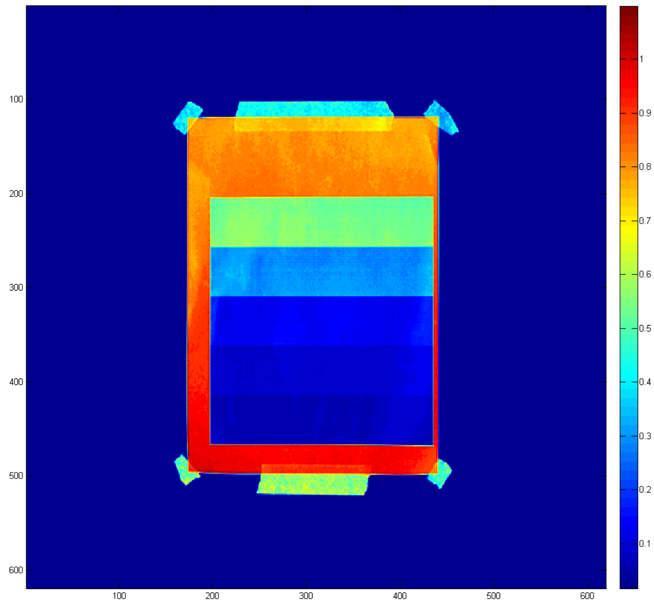

Figure 12. Nikon D800e Luminance Values Exported from Matlab $(0 - 1.018\frac{\text{cd}}{\text{m}^2})$.

5.2.7 Accuracy and error analysis of Nikon D800e

The current script is capable to present luminance data with median relative accuracy of 9.3%. This is calculated by comparing the relative difference of each point in the luminance data matrix from LMK LabSoft and the luminance data matrix created from Nikon D800e image. The relative difference between the two luminance values is calculated with the following formula:

$$R_v = (\frac{|Y_c - L_c|}{L_c}) \tag{6}$$

Where R_v is the relative difference value, Y_c is the luminance value of Nikon D800e matrix and L_c is the luminance value generated by TechnoTeam LMK LabSoft. The relative difference values compose a matrix of the same size as the experiment data. The median value of this matrix is 9.3%.

5.3 Errors in luminance measurements

Literature defines guidelines to the relative error in luminance levels the luminance measurement equipment should perform [2]. The following error tolerances in outdoor measurements are suggested:

- Road surface luminance: less than 10%

- Point luminance: less than 15%
- Veiling luminance: less than 25%

Error that comes from mounting the luminance photometer in a car is practically inevitable. The photometer should be mounted in the same place as the drivers eyes which is impossible. Therefore a place very close to the source of drivers eyes should be chosen. If possible the photometer should be mounted on the same rig as the eye tracker. Thus the shaking of the test environment such as car would effect both meters in the same manner. If this is not possible then the photometer can be mounted on the dashboard or on the roof.

One possible case would be a miniature luminance camera worn by the driver. Certain eye trackers such as SMI Eye Tracking Glasses reviewed before have this feature. It could be meaningful to test the SMI forward camera for its luminance measurement capabilities.

6 Mobile Mapping Systems

Mobile mapping means measuring and modelling the environment using mobile mapping devices. These devices may be laser scanners or traditional cameras and are made mobile most often by mounting them on a car. An important part is to get an accurate information about the mobile mapping device position and this is often accomplished by combined data of global positioning system (GPS) and inertia measurement unit (IMU). A well known example of mobile mapped environment model is the Google Maps Street View.

There are several reasons why combining eye tracking data to luminance data would benefit from mobile mapping systems. The existing mobile mappings and environment models can be useful when finding a suitable experiment area. Also the IMU and GPS position information common in mobile mapping systems will be extremely valuable when combining and synchronising the luminance data to eye fixation data. Later the luminance images can be textured to environment models to create a model of street lighting and environment under night-time conditions.

6.1 Combining luminance information to mobile mapped environment models

There are plenty of existing mobile mapped environment models such as Google Maps mentioned earlier. Built Environment Measurement and Modelling Institute of Aalto University also has mobile mapped environment models and are capable to map new areas under research.

RYMM has existing algorithms and expertise to add images of the environment as texture to laser mapped models. Texturing the environment with images containing the luminance information would not in practice differ from the texturing of images RYMM has made earlier. After executing laser mapping, luminance photometric measurements and model image texturing, an environment model textured with luminance images out of any chosen area can be generated. An example of environment model of river bank made by RYMM is presented in Figure 13.

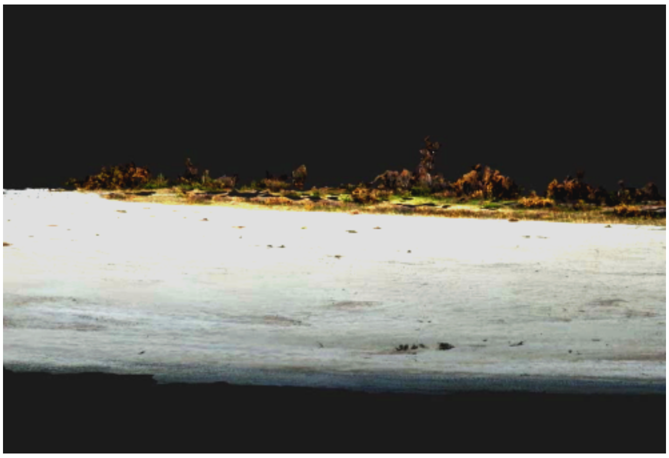

Figure 13. Example Picture of Mobile Mapped River Bank (Copyright: Matti Kurkela / Aalto-yliopisto, Geodeettinen Laitos)

7 Integrating eye tracking data to video luminance measurements

A demonstrative Matlab program that combines eye tracking data and video luminance mapping was created. The eye tracking data was exported from SMI BeGaze analysis software and the video material was the one recorded in pilot field measurements with Nikon D800e. The eye tracking data was generic example data and had absolutely no correspondence with the night time video measurements. Also the video luminance data is not calibrated, only converted to luminance values and pseudo-coloured. Thus the program created is completely demonstrative with no functionality for measurement purposes. The program created can be found in Appendix A. The example video can be found in the Light Energy project site:

http://www.lightinglab.fi/LightEnergy/private/lumi.avi

This linked video is only accessible by people in the Light Energy project, but it can be available by request for others also.

The performance of Matlab in this task is rather disappointing. To combine 3000 frames or 100s of video footage to eye tracking data the program took over 7 hours and 20 minutes. That means nearly half a minute for each second of data. The program has only functions for reading the data, interpreting RGB values as luminance values, transforming the luminance data to pseudo-colors, drawing the eye tracking data to video and exporting the data as '.avi' -video. The function where the adjusting of these two datasets for integration takes part and function for luminance data calibration are not implemented. It should be studied whether other programming environments such as Python, Java or C would be more efficient. Figure 14 presents the Matlab timer for the program performance.

Profile Summary
Generated 05-Jul-2013 19:01:20 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
pseudoc6	1	26430.388 s	25986.200 s	
...er.VideoWriter>VideoWriter.writeVideo	3000	355.110 s	1.579 s	
...oWriter>VideoWriter.writeStructFrames	3000	352.506 s	2.699 s	
...eoWriter>VideoWriter.writeImageFrames	3000	320.257 s	27.697 s	
IProfile>IProfile.writeVideoFrame	3000	287.026 s	0.263 s	
...tionJpegAviFilePlugin.writeVideoFrame	3000	286.763 s	0.304 s	
...ePlugin>AviFilePlugin.writeVideoFrame	3000	286.459 s	0.553 s	
OutputStream>OutputStream.write	3000	285.552 s	0.584 s	
OutputStream>OutputStream.writePackets	3000	284.968 s	282.144 s	
darkline	14700	88.961 s	88.961 s	
...oWriter>VideoWriter.convertColorspace	6000	30.573 s	1.224 s	
...Writer>VideoWriter.convertFramesToRGB	6000	29.050 s	28.213 s	
ismember	9000	3.548 s	0.743 s	
ismember>ismemberlegacy	9000	2.954 s	0.971 s	
cell.ismember	6000	2.842 s	0.323 s	
...bj.onCustomEvent(data.Type,data.Data)	3000	2.824 s	0.383 s	
cell.ismember>cellismemberlegacy	6000	2.519 s	0.952 s	

Figure 14. Timer data of the program performance

There are few aspects that must be particularly taken into account when creating the functional integration of eye tracking and video luminance data. Most probably the resolution will not be exactly same in both cases. Thus one of these data must be scaled to match the other. When scaling the data, proper spatial interpolation will be a necessity to preserve the accuracy of the data. Also the frame rate of these two data may be different. In this case the interpolation must be done in temporal manner. Interpolation or decisive down-sampling over systematic or random down-sampling is suggestible. The angle velocity of human eye can reach 900° degrees per second [8]. Then in 30fps eye can move 30° degrees within one frame which is very significant. It is therefore important to find and collect the large movements from the source data and not just accidentally ignore them with systematic algorithm. A very simple solution to down sample eye tracking data might be to remove the smallest changes in pairs or groups and interpolate between them.

To achieve spatial accuracy, focusing must be done properly. The eye tracker should always record not only eye movement video but the video of the scenery forward also. This way the data can be synchronized even if there was a problem in the time signature.

8 Conclusions

The optimal measurement equipment obviously depends strongly on the case studied. When high sample rate measurement data is desired one may have to give up a certain amount of luminance accuracy and dynamics. In the case of combining luminance measurements to eye tracking data high frame rate luminance data is needed. The researchers of a study where eye tracking and luminance photometry were combined encountered this problem [21]. The lack of high sample rate luminance data made the combined data analysis difficult.

The existing nearly contemporary market products such as LMK Mobile Advanced are very good in terms of usefulness and scientific accuracy. Yet they are not capable of recording high frame rate video. Thus we should continue studying the use of a video camera in luminance measurement. The research should be scalable so that we would not depend on some certain video camera model, but the whole variety of them including the future models.

The SMI Eye Tracking Glasses and the SmartEye are the best for eye tracking measurement. The SmartEye system is superior for permanent vehicle installation. It is quite expensive though, and collaboration within the science community is suggested if an existing installation could be available that way. SMI Eye Tracking Glasses on the other hand are a very good mobile, easily accessible and convenient research device. It is not as accurate as the SmartEye, but can be adjusted and applied to great variety of experiment situations easily.

One potential idea would be also to study how accurate luminance measurements could be made with the forward scenery cameras of the eye trackers. If considerable accuracy for particular research questions could be achieved, this would be a superior solution in terms of synergy and convenience.

In combining the luminance data to mobile mapped environment models expertise of environment modellers should be trusted. Naturally this should be done in a close dialogue with the ones providing the luminance data images. For synergy it would be ideal to make the luminance measurements and environment mapping simultaneously. If simultaneous mapping and measurement is not possible, the integration can be done later with the use of positioning data.

To integrate all of the different data (eye tracking data, luminance data and environment models), some amount of programming is demanded. The programming for user interface of data integration and analysis was done with Matlab in this thesis. It would we wise to look onto another alternatives such as Python or C to make the analysis more efficient.

9 Summary

In this thesis a suitable equipment to combine eye tracking and luminance measurements was examined. Different eye trackers and imaging luminance photometers were evaluated and compared. Two different systems for two different research situations were suggested. An idea of creating a high sample rate luminance photometer was also introduced. Piloting experiments to start this project were done in laboratory and in field. A rudimentary program to execute the combination of eye tracking and high sample rate luminance video was created. Guidelines to successfully calibrate this new system in the future were sketched. Next phases for the system improvement were designed. A demonstrative video of the new systems design and potential was made also. Study continues for a system that combines eye tracking to high frequency luminance data and integrates it to mobile mapped environment models.

References

[1] Peter R. Boyce, Human Factors In Lighting, 2nd Edition, Taylor & Francis Group, 2003, ISBN 0-7484-0949-1 (hbk)

[2] Duco Schreuder, Outdoor Lighting: Physics, Vision and Perception, Springer Science + Business Media B.V. 2008 ISBN: 978-1-4020-8601-4

[3] CIE 191:2010, Recommended System for Mesopic Photometry Based on Visual Performance, Comission Internationale DE L'Eclairage CIE Central Bureau, 2010, ISBN 978 3 901906 88 6

[4] Karen K de Valois, Russell L De Valois, Spatial Vision, Oxford University Press, USA, 1988, ISBN: 0195050193

[5] J.A. Jones, J. Edward Swan, M. Bolas, Peripheral Stimulation and its Effect on Perceived Spatial Scale in Virtual Environments, IEEE Transactions on Visualization and Computer Graphics, Scientific and Technical Information Division, April 2013

[6] Egil Haug, Olav Sand, Oeysten V. Sjaadtad, Ihmisen Fysiologia, WSOY, 2007, ISBN 978-951-0-19882-7

[7] Andrew T. Duchowski, Eye Tracking Methodology, Theory and Practice, Springer-Verlag London Limited, 2003, ISBN 1852336668

[8] Fischer, B. & Ramsperger, E. "Human express saccades: Extremely short reaction times of goal directed eye movements"., Experimental Brain Research 57., 1984

[9] Xindian Long, Eye Movement Tracking for Diagnostic Systems, bimagicLab, Center for Bioimage Informatics, Dept. of Electrical & Computer Engineering, Carnegie Mellon University, 2010

[10] Jan Winter and Stephan Völker, Typical Eye Fixation Areas of Car Drivers in Inner-City Environments at Night, Fachgebiet Lichttechnik, Technische Universität Berlin, 2013, March 18

[11] Sensomotoric Instruments Eye Tracking Glasses data sheet
http://www.eyetracking-glasses.com/fileadmin/user_upload/documents/smi_etg20_flyer.pdf, accessed 15.10.2013

[12] Sensomotoric Instruments iView X HED data sheet
http://www.smivision.com/fileadmin/user_upload/downloads/product_flyer/prod_smi_iviewx_hed.pdf, accessed 15.10.2013

[13] SmartEye Smart Eye Pro data sheet
http://www.smarteye.se/sites/smarteye/files/datasheets/smart_eye_pro.pdf, accessed 15.10.2013

[14] Applied Science Laboratories Mobile Eye-XG data sheet
http://www.asleyetracking.com/Site/Portals/0/Mobile%20Eye%20XG%20wireless.pdf, accessed 15.10.2013

[15] Tobii Glasses data sheet
http://www.tobii.com/Global/Analysis/Downloads/Product_Descriptions/Tobii_System_Recommendations_2.0.7.pdf?epslanguage=en, accessed 15.10.2013

[16] Liisa Halonen & Jorma Lehtovaara, Valaistustekniikka, Otatieto Oy, Gummerus Kirjapaino, 1992, ISBN: 951-672-145-1

[17] TechnoTeam LMK Mobile data sheet
http://www.technoteam.de/e898/e97/e286/e287/e1028/LMK_ma_EOS550D_EN_eng.pdf , accessed 15.10.2013

[18] LMK brochure
http://www.technoteam.de/products/lmk_luminance_measuring_camera/lmk_5/index_eng.html , accessed 15.10.2013

[19] Radiant Imaging ProMetric Version 6.0.6 User's Manual
Radiant Imaging, Inc
2002

[20] Instument Systems Lumicam 1300 data sheet
http://www.instrumentsystems.com/fileadmin/editors/downloads/Products/LumiCam_1300_e.pdf , accessed 15.10.2013

[21] Can Cengiz MSc, Henri Kotkanen BA, Marjukka Puolakka DSc, Otto Lappi MA2, Esko Lehtonen MA2, Liisa Halonen DSc, Heikki Summala PhD2
Corresponding author: Can Cengiz, Department of Electronics, Lighting Unit, Aalto University School of Electrical Engineering
Combined eye-tracking and luminance measurements when driving on a rural road: towards determining of mesopic adaptation luminance when

driving

[22] Nikon D800/D800E Manual http://www.nikonusa.com/pdf/manuals/dslr/D800_EN.pdf, accessed 15.10.2013

[23] ITU-R BT.601-7, Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, Electronic Publication, Geneva, 2011

[24] Deane B. Judd, The 1931 I. C. I. Standard Observer and Coordinate System for Colorimetry, Bureau of Standards, Washington, D. C., 1933

[25] John W.T. Walsh, Photometry, Constable and Company, Ltd., 1965

A Appendix A, Matlab Code for Nikon D800e Luminance Data Analysis and Presentation

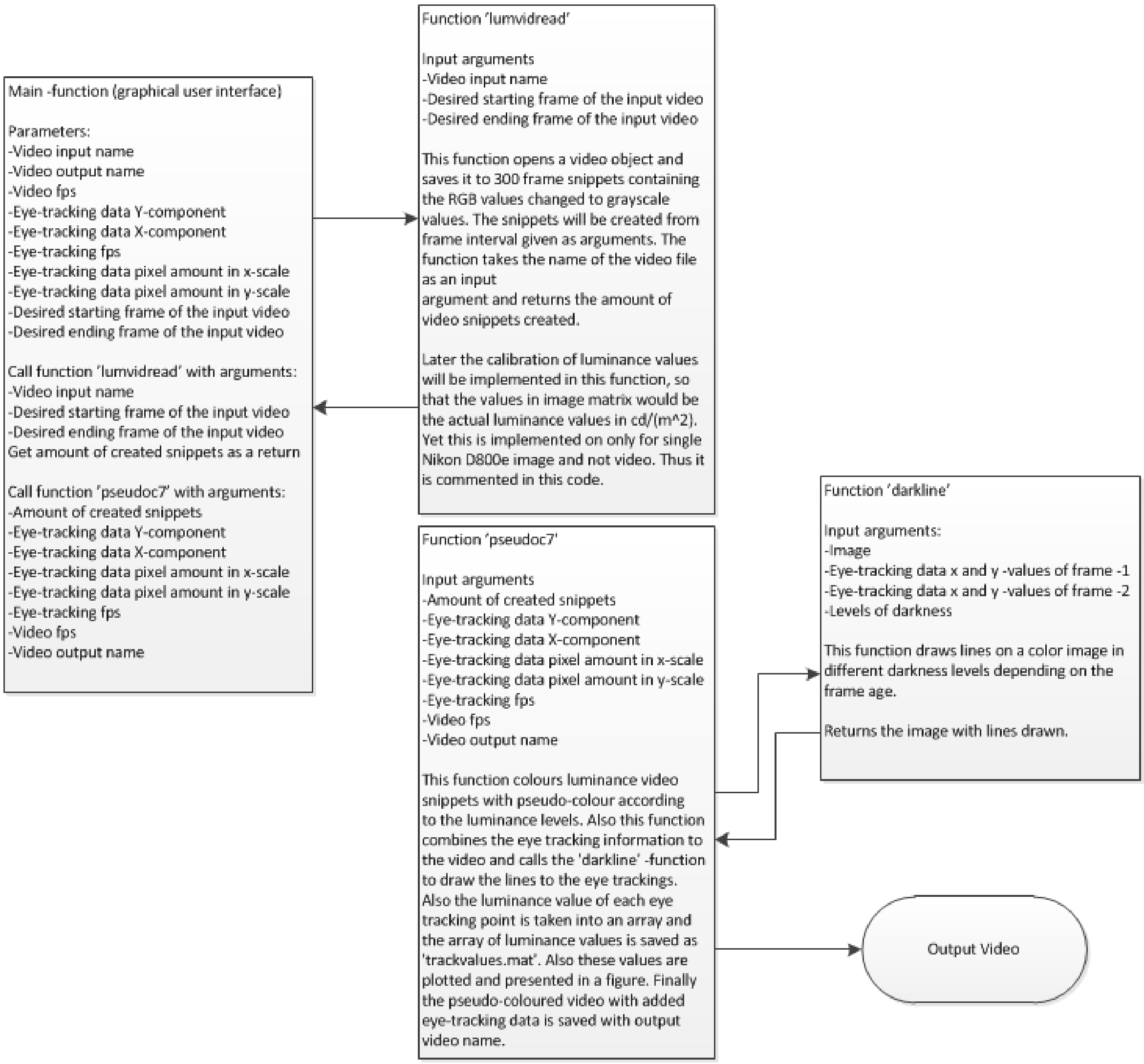

Figure 15. Matlab Code Flowchart.

```matlab
%%
function lumimaingui(inputname,outputname,videofps,
    eyedatax,eyedatay,eyedatafps,eyescalex,eyescaley,vsp,
    vep)
%This funcion is the main funcion to control the GUI.

snips = lumvidread(inputname,vsp,vep);

startpoint = 1;
```

```

endpoint = snips;

pseudoc7(startpoint,endpoint, eyedatax, eyedatay, eyescalex, eyescaley, eyedatafps, videofps, outputname)

end
%%

%%
function varargout = lumigui(varargin)
% LUMIGUI MATLAB code for lumigui.fig
%      LUMIGUI, by itself, creates a new LUMIGUI or raises the existing
%      singleton*.
%
%      H = LUMIGUI returns the handle to a new LUMIGUI or the handle to
%      the existing singleton*.
%
%      LUMIGUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in LUMIGUI.M with the given input arguments.
%
%      LUMIGUI('Property','Value',...) creates a new LUMIGUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before lumigui_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to lumigui_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help lumigui
```

```
% Last Modified by GUIDE v2.5 23-Jul-2013 13:14:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @lumigui_OpeningFcn, ...
                   'gui_OutputFcn',  @lumigui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before lumigui is made visible.
function lumigui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to lumigui (see VARARGIN)

% Choose default command line output for lumigui
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```
% UIWAIT makes lumigui wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = lumigui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function inname_Callback(hObject, eventdata, handles)
% hObject    handle to inname (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of inname as text
%        str2double(get(hObject,'String')) returns contents of inname as a double


% --- Executes during object creation, after setting all properties.
function inname_CreateFcn(hObject, eventdata, handles)
% hObject    handle to inname (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```
% Hint: edit controls usually have a white background on
  Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
  'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function outname_Callback(hObject, eventdata, handles)
% hObject    handle to outname (see GCBO)
% eventdata  reserved - to be defined in a future version
   of MATLAB
% handles    structure with handles and user data (see
  GUIDATA)

% Hints: get(hObject,'String') returns contents of
  outname as text
%        str2double(get(hObject,'String')) returns
  contents of outname as a double


% --- Executes during object creation, after setting all
  properties.
function outname_CreateFcn(hObject, eventdata, handles)
% hObject    handle to outname (see GCBO)
% eventdata  reserved - to be defined in a future version
   of MATLAB
% handles    empty - handles not created until after all
  CreateFcns called

% Hint: edit controls usually have a white background on
  Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
  'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function vidfps_Callback(hObject, eventdata, handles)
% hObject    handle to vidfps (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vidfps as text
%        str2double(get(hObject,'String')) returns contents of vidfps as a double


% --- Executes during object creation, after setting all properties.
function vidfps_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vidfps (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function eyex_Callback(hObject, eventdata, handles)
% hObject    handle to eyex (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of eyex as text
%        str2double(get(hObject,'String')) returns contents of eyex as a double


% --- Executes during object creation, after setting all properties.
```

```
function eyex_CreateFcn(hObject, eventdata, handles)
% hObject    handle to eyex (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    empty - handles not created until after all
   CreateFcns called

% Hint: edit controls usually have a white background on
   Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
   'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function eyey_Callback(hObject, eventdata, handles)
% hObject    handle to eyey (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
   GUIDATA)

% Hints: get(hObject,'String') returns contents of eyey
   as text
%        str2double(get(hObject,'String')) returns
   contents of eyey as a double


% --- Executes during object creation, after setting all
   properties.
function eyey_CreateFcn(hObject, eventdata, handles)
% hObject    handle to eyey (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    empty - handles not created until after all
   CreateFcns called

% Hint: edit controls usually have a white background on
   Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
   'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end



function eyefps_Callback(hObject, eventdata, handles)

% hObject    handle to eyefps (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of eyefps as text
%        str2double(get(hObject,'String')) returns contents of eyefps as a double


% --- Executes during object creation, after setting all properties.
function eyefps_CreateFcn(hObject, eventdata, handles)
% hObject    handle to eyefps (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
inputname = get(handles.inname,'String');
outputname = get(handles.outname,'String');
videofps = str2num(get(handles.vidfps,'String'));
eyedatax = get(handles.eyex,'String');
eyedatay = get(handles.eyey,'String');
eyedatafps = str2num(get(handles.eyefps,'String'));
```

```
eyescalex = str2num(get(handles.scalex,'String'));
eyescaley = str2num(get(handles.scaley,'String'));
startpoint = str2num(get(handles.vsp,'String'));
endpoint = str2num(get(handles.vep,'String'));
lumimaingui(inputname,outputname,videofps,eyedatax,
    eyedatay,eyedatafps,eyescalex,eyescaley,startpoint,
    endpoint);

%guidata(hObject,handles);
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version
     of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles
    )
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version
     of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)



function scalex_Callback(hObject, eventdata, handles)
% hObject    handle to scalex (see GCBO)
% eventdata  reserved - to be defined in a future version
     of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

% Hints: get(hObject,'String') returns contents of scalex
     as text
%        str2double(get(hObject,'String')) returns
    contents of scalex as a double


% --- Executes during object creation, after setting all
    properties.
function scalex_CreateFcn(hObject, eventdata, handles)
% hObject    handle to scalex (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    empty - handles not created until after all
   CreateFcns called

% Hint: edit controls usually have a white background on
   Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
   'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function scaley_Callback(hObject, eventdata, handles)
% hObject    handle to scaley (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
   GUIDATA)

% Hints: get(hObject,'String') returns contents of scaley
    as text
%        str2double(get(hObject,'String')) returns
   contents of scaley as a double


% --- Executes during object creation, after setting all
   properties.
function scaley_CreateFcn(hObject, eventdata, handles)
% hObject    handle to scaley (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    empty - handles not created until after all
   CreateFcns called

% Hint: edit controls usually have a white background on
   Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
   'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function vsp_Callback(hObject, eventdata, handles)
% hObject    handle to vsp (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

% Hints: get(hObject,'String') returns contents of vsp as
    text
%        str2double(get(hObject,'String')) returns
    contents of vsp as a double


% --- Executes during object creation, after setting all
    properties.
function vsp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vsp (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    empty - handles not created until after all
    CreateFcns called

% Hint: edit controls usually have a white background on
    Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
    'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function vep_Callback(hObject, eventdata, handles)
% hObject    handle to vep (see GCBO)
% eventdata  reserved - to be defined in a future version
    of MATLAB
% handles    structure with handles and user data (see
    GUIDATA)

% Hints: get(hObject,'String') returns contents of vep as
    text
%        str2double(get(hObject,'String')) returns
    contents of vep as a double
```

```


% --- Executes during object creation, after setting all
    properties.
function vep_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vep (see GCBO)
% eventdata  reserved - to be defined in a future version
     of MATLAB
% handles    empty - handles not created until after all
    CreateFcns called

% Hint: edit controls usually have a white background on
    Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,
    'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%%

%%
function snips = lumvidread( vidname,vsp,vep )
%This function opens a video object
%and saves it to 300 frame snippets containing the RGB
    values changed
%to grayscale values. The snippets will be created from
%frames given as arguments. The 'vsp' input argument
    indigates the
%start frame and 'vep' the end frame. The last snippet
%will contain the remaing frames.
%The snippets will be named "snippet-(running number).mat
    ".
%The function takes the name of the video file as an
    input
%argument and returns the amount of video snippets
    created.
%Later the calibration of luminance values will be
    implemented in this
%function, so that the values in image matrix would be
    the actual luminance
%values in cd/(m^2). Yet this is implemented on only for
    single Nikon D800e image and not video. Thus it is
    commented in this code.

vidobj = VideoReader(vidname);
```

```
frames_all = get(vidobj, 'NumberOfFrames');
frames = vep-vsp;
modulus = mod(frames,300);
rounds = floor(frames/300);

b = 1;
c = vsp;
d = 1;

while b <= rounds
    for k = c : c+299
        lumvid{d} = read(vidobj,k);
        d = d+1;
    end
    c = c+300;
    d=1;
    filename = strcat('snippet-',num2str(b),'.mat');
    save(filename,'lumvid');
    clear lumvid;
    b = b+1;
end

for k = c : (c+modulus-1)
    lumvid{d} = read(vidobj,k);
    d = d + 1;
end

filename = strcat('snippet-',num2str(b),'.mat');
save(filename,'lumvid');
clear lumvid;
clear vidobj;

%RGB to luminance conversion
for l = 1:rounds+1
    filename = strcat('snippet-',num2str(l),'.mat');
    load(filename);
    counter = 1;
    while iscell(lumvid{counter})
       counter = counter+1;
    end
    sizetemp = lumvid{1};
    [x,y,z]=size(sizetemp);
    clear sizetemp;
    ksk = numel(lumvid);
```

```
    for k = 1:ksk
        imagetemp = lumvid{k};
        for i = 1 : y
            for j = 1 : x
                imagetempl(j,i) =((imagetemp(j,i,1)*0.3+
                   imagetemp(j,i,2)*0.59+imagetemp(j,i,3)
                   *0.11));
            end
        end
%Commented calibration section
%       for i = 1 : y
%           for j = 1 : x
%               imagetempl_cal(j,i) =1.016^(imagetempl(j,
   i));
%           end
%       end
%       maximtlc=max(imagetempl_cal(:));
%       scalecoeff = 1.098/maximtlc;
%       imagetempl_cal = imagetempl_cal*scalecoeff;
%        for i = 1 : 619
%           for j = 1 : 619
%               if imagetempl_cal(j,i) < 0.05
%                       imagetempl_cal(j,i) = (
   imagetempl_cal(j,i)^1.016)*0.58;
%               end
%           end
%       end
%        lumvid{k} = imagetempl_cal;

        lumvid{k} = imagetempl;
    end
    savename = strcat('lumsnippet-',num2str(l),'.mat');
    save(savename,'lumvid');
    clear lumvid;
end
snips = rounds+1;
end
%%

%%
function pseudoc7(startpoint, endpoint, smidatabx,
   smidataby, eyescalex, eyescaley, eyedatafps, videofps,
    outputname)
%This function colours luminance video snippets with
   pseudo-colour according
```

```
%to the luminance levels. Also this function combines the
    eye tracking
%information to the video and calls the 'darkline' -
    function to draw the
%lines to the eye trackings. Also the luminance value of
    each eye
%tracking point is taken into an array and the array of
    luminance values is
%saved as 'trackvalues.mat'. Also these values are
    plotted and presented in
%a figure.


writerObj = VideoWriter(outputname);
open(writerObj);
load(smidatabx)
load(smidataby)

for kak = startpoint:endpoint
    filename = strcat('lumsnippet-',num2str(kak),'.mat');
    load(filename);
    lum = lumvid{1};
    [p,q,r] = size(lum);

    xsca = q/eyescalex;
    ysca = p/eyescaley;

    Xs=smidatabx*xsca;
    Ys=smidataby*ysca;
    Xs= floor(Xs);
    Ys= floor(Ys);

    frames = numel(lumvid);

    for k = 1:frames

        lum = lumvid{k};
        lum_temp = lumvid{k};
        lum = double(lum);
        lummax = max(lum(:));
        lumsc = 765/lummax;
        floor(lumsc);
        lum = lum*lumsc;
        lum = uint16(lum);
        [p,q,r] = size(lum);
```

```
        plum = ones(p,q,3);
        plum = uint8(plum);

        tt=Ys(k);
        rr=Xs(k);
        trackvalues(k) = lum_temp(tt,rr);

        for i=1:1:p
            for j=1:1:q
                val = lum(i,j);
                if (val>= 0) && (val< 255)
                    plum(i,j,1)=0;
                    plum(i,j,2)=0;
                    plum(i,j,3)=0+val;

                elseif (val>= 255) && (val< 510)
                    plum(i,j,1)=0;
                    plum(i,j,2)=0+(val-255)*2;
                    plum(i,j,3)=255-(val-255);

                elseif (val>= 510) && (val< 766)
                    plum(i,j,1)=0+(val-510);
                    plum(i,j,2)=255-(val-510);
                    plum(i,j,3)=0;

                end
            end
        end

    for i = Xs(k)-5 : Xs(k)+5
            for j = Ys(k) : Ys(k)+1
                    plum(j,i,1) = 255;
                    plum(j,i,2) = 255;
                    plum(j,i,3) = 255;

                end
            end
        for i = Xs(k) : Xs(k)+1

            for j = Ys(k)-5 : Ys(k)+5
                plum(j,i,1) = 255;
                plum(j,i,2) = 255;
                plum(j,i,3) = 255;

            end
```

```
        end

    if k > 6

        for m = 1:5
            plum = darkline(plum,Xs(k-m),Ys(k-m),Xs(k-m+1),Ys(k-m+1), m);
        end

    end

    frame = im2frame(plum);
    writeVideo(writerObj,frame);
    end

end
close(writerObj)
save('trackvalues.mat','trackvalues');
figure(2);
plot(trackvalues);
end
%%

%%
function M = darkline(I, p1x, p1y, p2x, p2y, dark)
% This function draws lines on a color image in 5
% different darkness levels depending on the frame age,
% where p1x(i), p1y(i): first point (x,y) and
% p2x(i), p2y(i): second point (x,y)
% and returns M, the image matrix with lines drawn.
% Modified by -/\/\-/\/\- from imline.m ( Ethan Png )

len = length(p1x);
for i=1:len
   if p1x(i) > p2x(i)
      px1 = round(p2x(i)); py1 = round(p2y(i));
      px2 = round(p1x(i)); py2 = round(p1y(i));
   else
      px1 = round(p1x(i)); py1 = round(p1y(i));
      px2 = round(p2x(i)); py2 = round(p2y(i));
   end
   if p1y(i) > p2y(i)
      ty1 = round(p2y(i)); tx1 = round(p2x(i));
      ty2 = round(p1y(i)); tx2 = round(p1x(i));
   else
```

```
        ty1 = round(p1y(i)); tx1 = round(p1x(i));
        ty2 = round(p2y(i)); tx2 = round(p2x(i));
    end
    h = py2 - py1;
    w = px2 - px1;
    if w ~= 0
        for p=px1:px2
            y = round((h*(p-px1)/w)+py1);
            if dark == 1
                I(y,p,1) = 255;
                I(y,p,2) = 255;
                I(y,p,3) = 255;
            end
            if dark == 2
                I(y,p,1) = 200;
                I(y,p,2) = 200;
                I(y,p,3) = 200;
            end
            if dark == 3
                I(y,p,1) = 150;
                I(y,p,2) = 150;
                I(y,p,3) = 150;
            end
            if dark == 4
                I(y,p,1) = 100;
                I(y,p,2) = 100;
                I(y,p,3) = 100;
            end
            if dark == 5
                I(y,p,1) = 1;
                I(y,p,2) = 1;
                I(y,p,3) = 1;
            end
        end
    end
    h = ty2 - ty1;
    w = tx2 - tx1;
    if h ~= 0
        for t=ty1:ty2
            x = round((w*(t-ty1)/h)+tx1);

                if dark == 1
                    I(t,x,1) = 255;
                    I(t,x,2) = 255;
                    I(t,x,3) = 255;
```

```
            end
            if dark == 2
                I(t,x,1) = 200;
                I(t,x,2) = 200;
                I(t,x,3) = 200;
            end
            if dark == 3
                I(t,x,1) = 150;
                I(t,x,2) = 150;
                I(t,x,3) = 150;
            end
            if dark == 4
                I(t,x,1) = 100;
                I(t,x,2) = 100;
                I(t,x,3) = 100;
            end
            if dark == 5
                I(t,x,1) = 1;
                I(t,x,2) = 1;
                I(t,x,3) = 1;
            end


      end
   end
end
M = I;
%%
```